

REMARKS

Entry of this amendment after final is respectfully requested as placing this case in condition for allowance or better form for appeal. Except for Claims 8, 12, 13 and 31, 35, 36, all finally rejected claims are being canceled by this amendment. As to Claims 8, 12, 13 and 31, 35, 36, dependent Claims 8 and 31 are merely being amended to be in independent form and are not being substantively amended (but rather are merely being amended as to form). Claims 12, 13 and 35, 36 depend upon these amended Claims 8 and 31, respectively. Finally, this amendment after final amends objected to, but allowable, claims to place them in condition for allowance.

Upon entry of this amendment after final, Claims 8, 12, 13, 18-23, 31, 35, 36, 41 and 46 will be pending in the present application. Claims 8, 18-23, 31, 41 and 46 are being amended, and Claims 1-7, 9-11, 14-17, 24-30, 32-34, 37-40, 42-45 and 47-50 are being canceled, herewith. Reconsideration of the pending claims is respectfully requested.

I. Requested Copy of Referenced Art

The Examiner requested copies of documentation in connection with the NICE mechanism referred to by Applicants at pp 20-21 of the present application. Applicants urge that as mentioned on page 21, line 5-7, NICE is a standard UNIX dispatch priority mechanism. Applicants are including herewith in Appendix A a list of standard UNIX commands (UNIX being a computer operating system known to those of skill in the art), which includes of listing of such NICE command. In addition, Applicants are including herewith in Appendix B the command syntax for invoking the NICE command under UNIX. Because this is a standard UNIX command commonly known to those of skill in the art, no further information is being provided herewith in response to the Examiner's documentation request.

II. Obviousness-type Double Patenting Rejection

The Examiner rejected Claim 1 under the judicially created doctrine of obviousness-type double patenting. Applicants are canceling Claim 1 herewith without prejudice or disclaimer, and thus this rejection is now moot.

III. 35 U.S.C. § 103, Obviousness

The Examiner rejected Claims 1-17, 24-40 and 47-50 under 35 U.S.C. § 103 as being unpatentable over Ferguson et al. (U.S. Patent 5,504,894) in view of Vaitzblit et al. (U.S. Patent 5,528,513). This rejection is respectfully traversed.

With respect to Claims 1-7, such claims have been canceled herewith, without prejudice or disclaimer.

With respect to Claim 8, such claim has merely been amended to be in independent form to include all features previously recited in Claim 1, of which Claim 8 depended upon. In rejecting Claim 1, the Examiner cited Ferguson as reading on the workload management with respect to classes in other tiers based on priorities of the tiers (citing Ferguson p. 3 17-20), and cited Vaitzblit as reading on the claimed workload management with respect to other classes within the same tier (citing Vaitzblit p.4 40-55). In rejecting Claim 8, which is a further refinement of the workload management with respect to other classes within the same tier, the Examiner cites Ferguson p3 39-47. Applicants show error in this rejection of Claim 8 as follows.

Claim 8 recites the feature of “wherein performing workload management with respect to other classes within a same tier comprises determining a percentage goal for the process as a function of a number of system resource shares associated with the class in which the process is classified divided by a total number of shares allocated to active classes in the same tier as the class in which the process is classified”. In other words, determining a percentage goal for the process is determined as a function of:

(# of system resource shares associated with this class) divided by (total number of shares allocated to active classes in this tier)

Thus, for example, if a class has 6 shares and there are 10 total shares within the active classes of its tier, the class percentage goal is 60% of the system resources (Specification page 17, lines 10-12).

The passage cited by the Examiner in rejecting Claim 8 states that the workload manager priority rates or orders the classes in accordance with the current class

performance indices such that a transaction of a class that is performing poorly gets a higher dispatch priority than a transaction of a class that is performing better. The current class performance indices that are used by the workload manager in this determination are “a ratio of the current average class response time and the class response time goal” (Ferguson p3 40-43). In other words:

performance index = (average class response time) divided by (class response time goal)

This is also stated by Ferguson at p5 57-64, and described in detail at Ferguson p7 15-59. If this performance index is ≤ 1 , the class response time goal is being met, whereas if the performance index is > 1 , the class response time goal is not being met. Thus, Ferguson teaches that a given transaction for a class is routed according to its class’ performance index, and transactions for classes performing more poorly against their performance goals receive higher dispatch priority than transactions for other classes that are performing better with respect to their response time goals. Thus, the prioritization of transaction dispatch is determined by actual response time (for a class) versus goal response time (for a class), in an attempt to increase the likelihood that all classes will obtain their respective response time goals. It should be noted that *these response time goals are set by a data base administrator* (Ferguson p5 24-26). This is in contrast to what is recited in Claim 8, which determines a percentage goal for a process based on two actual system parameters: (i) the number of system resource shares associated with this class, and (ii) the total number of allocated shares for all active classes in this tier. These two parameters are used to determine a percentage goal for a process, and thus this process is very different from a database administrator *assigning* response time goals for a transaction class, with no description of how such goals are obtained or determined.

Thus, it is shown that the Examiner has failed to establish a *prima facie* showing of obviousness with respect to Claim 8. As such, the burden has not shifted to Applicants to rebut an obviousness assertion, and Claim 8 is shown to have been erroneously rejected¹.

¹ In rejecting claims under 35 U.S.C. Section 103, the examiner bears the initial burden of presenting a *prima facie* case of obviousness. *In re Oetiker*, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992). To establish *prima facie* obviousness of a claimed invention, all of the claim limitations must be

With respect to Claims 9-11, such claims have been canceled herewith, without prejudice or disclaimer.

With respect to Claims 12 and 13, Applicants traverse for similar reasons to those given above regarding Claim 8 (of which Claims 12 and 13 depend upon).

With respect to Claims 14-17 and 24-30, such claims have been canceled herewith, without prejudice or disclaimer.

With respect to Claim 31, such claim has merely been amended to be in independent form to include all features previously recited in Claim 24, of which Claim 31 depended upon. Applicants traverse the rejection of Claim 31 for similar reasons to those given above regarding Claim 8.

With respect to Claims 32-34, such claims have been canceled herewith, without prejudice or disclaimer.

With respect to Claims 35 and 36, Applicants traverse for similar reasons to those given above regarding Claim 31 (of which Claims 35 and 36 depend upon).

With respect to Claims 37-40 and 47-50, such claims have been canceled herewith, without prejudice or disclaimer.

Therefore, the rejection of Claims 1-17, 24-40 and 47-50 under 35 U.S.C. § 103 has been overcome.

IV. Objection to Claims

The Examiner stated that Claims 18-23 and 41-46 were objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. In response, Claims 18-23, 41 and 46 have been rewritten accordingly to overcome this objection. Claims 42-45 have been canceled herewith, without prejudice or disclaimer,

taught or suggested by the prior art. MPEP 2143.03. *See also, In re Royka*, 490 F.2d 580 (C.C.P.A. 1974). Only if that burden is met, does the burden of coming forward with evidence or argument shift to the applicant. *Id. In re Bell*, 991 F.2d 781, 782, 26 USPQ2d 1529, 1531 (Fed. Cir. 1993) (quoting *In re Rinehart*, 531 F.2d 1048, 1051, 189 USPQ 143, 147 (CCPA 1976)). If the examiner fails to establish a *prima facie* case, the rejection is improper and will be overturned. *In re Fine*, 837 F.2d 1071, 1074, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988).

due to the large number of independent claims that would result if these claims were amended to be in independent form.

V. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: 7/27/04

Respectfully submitted,



Duke W. Yee
Reg. No. 34,285
Wayne P. Bailey
Reg. No. 34,289
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 367-2001
Attorneys for Applicants

Quick Reference List Of Unix Commands

General Commands

- **passwd** - change your password
- **exit, logout** - log out
- **man** - display Unix manual pages

File Handling

- **emacs, pico, vi** - editors
- **cat > file** - type in a file
- **cat file** - display a file
- **more, less** - display file a page at a time
- **ls** - list files
- **rm** - delete files
- **mv** - rename files
- **cp** - copy files
- **cd** - change directory
- **pwd** - show current directory
- **mkdir** - create directory
- **rmdir** - remove empty directory
- **chmod** - change access permissions on existing files
- **umask** - change default access permissions for new files

Printing

- **lpr** - print file
- **lpq** - list print jobs
- **cancel** - cancel print jobs

Email, WWW And Networking

- **elm, pine** - read and send email
- **netscape** - browse the World Wide Web
- **lynx** - browse the World Wide Web (text only)
- **ssh** - connect from one machine to another securely (recommended)
- **telnet, rlogin** - less secure ways to connect between machines (not recommended)
- **ftp** - transfer files between machines

Software

- **tex, latex** - compile a TeX or Latex file
- **xdvi** - view a DVI file
- **dvips** - convert a DVI file to Postscript

- **gv** - view a Postscript file
- **Splus6**
- **mathematica, math**

Process Management

- **ps** - list processes
- **top** - list all processes, starting with those taking the most CPU time
- **command &** - run command in the background
- **nice command &** - run command in the background at a reduced priority
- **kill** - kill a process

This document was written by the Statistical Laboratory Computer Officer, Eva Myers (eva@statslab.cam.ac.uk). It is available online at <http://www.statslab.cam.ac.uk/~eva/unixref.html>.

[\[Back to my official home page\]](#)

The Single UNIX® Specification, Version 2
Copyright © 1997 The Open Group

NAME

nice - invoke a utility with an altered system scheduling priority

SYNOPSIS

```
nice [-n increment] utility [argument...]
```

```
nice [-increment] utility [argument...]
```

DESCRIPTION

The *nice* utility invokes a utility, requesting that it be run with a different system scheduling priority (see the definition of **system scheduling priority** in the **XBD** specification, [Glossary](#)). With no options and only if the user has appropriate privileges, the executed utility is run with a system scheduling priority that is some implementation-dependent quantity less than or equal to the system scheduling priority of the current process. If the user lacks appropriate privileges to affect the system scheduling priority in the requested manner, the *nice* utility will not affect the system scheduling priority; in this case, a warning message may be written to standard error, but this will not prevent the invocation of *utility* or affect the exit status.

OPTIONS

The *nice* utility supports the **XBD** specification, [Utility Syntax Guidelines](#) except that the obsolescent version allows a multi-digit decimal integer as an option name.

The following option is supported:

-n increment
-increment

Specify how the system scheduling priority of the executed utility will be adjusted. The *increment* option-argument is a positive or negative decimal integer that will be used to modify the system scheduling priority of the executed utility in an implementation-dependent manner. Positive *increment* values cause a lower or unchanged system scheduling priority. Negative *increment* values may require appropriate privileges and will cause a higher or unchanged system scheduling priority. The system scheduling priority is bounded in an implementation-dependent manner. If the requested *increment* would raise or lower the system scheduling priority of the executed utility beyond implementation-dependent limits, then the limit whose value was exceeded is used.

OPERANDS

The following operands are supported:

utility The name of a utility that is to be invoked. If the *utility* operand names any of the special built-in utilities in Special Built-in Utilities, the results are undefined.

argument

Any string to be supplied as an argument when invoking the utility named by the *utility* operand.

STDIN

Not used.

INPUT FILES

None.

ENVIRONMENT VARIABLES

The following environment variables affect the execution of *nice*:

LANG

Provide a default value for the internationalisation variables that are unset or null. If *LANG* is unset or null, the corresponding value from the implementation-dependent default locale will be used. If any of the internationalisation variables contains an invalid setting, the utility will behave as if none of the variables had been defined.

LC_ALL

If set to a non-empty string value, override the values of all the other internationalisation variables.

LC_CTYPE

Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single- as opposed to multi-byte characters in arguments).

LC_MESSAGES

Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

NLSPATH

Determine the location of message catalogues for the processing of *LC_MESSAGES*.

PATH

Determine the search path used to locate the utility to be invoked. See the **XBD** specification, Environment Variables.

ASYNCHRONOUS EVENTS

Default.

STDOUT

Not used.

STDERR

Used only for diagnostic messages.

OUTPUT FILES

None.

EXTENDED DESCRIPTION

None.

EXIT STATUS

If the *utility* utility is invoked, the exit status of *nice* will be the exit status of *utility*; otherwise, the *nice* utility will exit with one of the following values:

- 1' An error occurred in the *nice* utility.
- 126 The utility specified by *utility* was found but could not be invoked.
- 127 The utility specified by *utility* could not be found.

CONSEQUENCES OF ERRORS

Default.

APPLICATION USAGE

Note that, in the obsolescent version, **-5** is a positive *increment*, while **--5** is a negative *increment*.

The only guaranteed portable uses of this utility are:

nice utility

Run *utility* with the default lower system scheduling priority.

nice -n <positive integer> utility

Run *utility* with a lower system scheduling priority.

On some systems they will have no discernible effect on the invoked utility and on some others they will be exactly equivalent.

Historical systems have frequently supported the *<positive integer>* up to 20. Since there is no error penalty associated with guessing a number that is too high, users without access to the system conformance document (to see what limits are actually in place) could use the historical 1 to 20 range or attempt to use very large numbers if the job should be truly low priority.

The system scheduling priority value of a process can be displayed using the command:

```
ps -o nice
```

The command, env, nice, nohup, time and xargs utilities have been specified to use exit code 127 if an error occurs so that applications can distinguish "failure to find a utility" from "invoked utility exited with an error indication". The value 127 was chosen because it is not commonly used for other meanings; most utilities use small values for "normal error conditions" and the values above 128 can be confused with termination due to receipt of a signal. The value 126 was chosen in a similar manner to indicate that the utility could be found, but not invoked. Some scripts produce meaningful error messages differentiating the 126 and 127 cases. The distinction between exit codes 126 and 127 is based on KornShell practice that uses 127 when all attempts to *exec* the utility fail with [ENOENT], and uses 126 when any attempt to *exec* the utility fails for any other reason.

EXAMPLES

None.

FUTURE DIRECTIONS

None.

SEE ALSO

[renice](#).

UNIX ® is a registered Trademark of The Open Group.

Copyright © 1997 The Open Group

[[Main Index](#) | [XSH](#) | [XCU](#) | [XBD](#) | [XCURSES](#) | [XNS](#)]
